

Techniques for effective and efficient fire detection from social media images

Marcos Bedo, Gustavo Blanco, Willian Oliveira, Mirela Cazzolato, Alceu Costa, Jose Rodrigues,
Agma Traina and Caetano Traina

Institute of Mathematics and Computer Science, University of São Paulo

Av. Trabalhador São Carlense, 400, São Carlos, SP, Brazil

{bedo, willian, alceufc, junio, agma, caetano}@icmc.usp.br, {blanco, mirelac}@usp.br

Keywords: Fire Detection, Feature Extraction, Evaluation Functions, Image Descriptors, Social Media

Abstract: Social media could provide valuable information to support decision making in crisis management, such as in accidents, explosions and fires. However, much of the data from social media are images, which are uploaded in a rate that makes it impossible for human beings to analyze them. Despite the many works on image analysis, there are no fire detection studies on social media. To fill this gap, we propose the use and evaluation of a broad set of content-based image retrieval and classification techniques for fire detection. Our main contributions are: (i) the development of the Fast-Fire Detection method (*FFireDt*), which combines feature extractor and evaluation functions to support instance-based learning; (ii) the construction of an annotated set of images with ground-truth depicting fire occurrences – the Flickr-Fire dataset; and (iii) the evaluation of 36 efficient image descriptors for fire detection. Using real data from Flickr, our results showed that *FFireDt* was able to achieve a precision for fire detection comparable to that of human annotators. Therefore, our work shall provide a solid basis for further developments on monitoring images from social media.

1 INTRODUCTION

Disasters in industrial plants, densely populated areas, or even crowded events may impact property, environment, and human life. For this reason, a fast response is essential to prevent or reduce injuries and financial losses, when crises situations strike. The management of such situations is a challenge that requires fast and effective decisions based on the best data available, because decisions based on incorrect or lack of information may cause more damage (Russo, 2013). One of the alternatives to improve information correctness and availability for decision making during crises is the use of software systems to support experts and rescue forces (Kudyba, 2014).

Systems aimed at supporting salvage and rescue teams often rely on images to understand the crisis scenario and to design the actions that will reduce losses. Crowdsourcing and social media, as massive sources of images, possess a great potential to assist such systems. Web sites such as Flickr, Twitter, and Facebook allow users to upload pictures from mobile devices, what generates a flow of images that carries valuable information. Such information may reduce the time spent to make decisions and it can be used

along with other information sources. Automatic image analysis is important to understand the dimensions, type, and the objects and people involved in an incident.

Despite the potential benefits, we observed that there is still a lack of studies concerning automatic content-based processing of crisis images (Villela et al., 2014). In the specific case of fire – which is observed during explosions, car accidents, forest and building fire, to name a few, there is an absence of studies to identify the most adequate content-based retrieval techniques (image descriptors) able to identify and retrieve relevant images captured during crises. In this work, we fill one of the gaps providing an architecture and an evaluation of the techniques for fire monitoring in images collected from social media.

This work reports on one of the steps of the project *Reliable and Smart Crowdsourcing Solution for Emergency and Crisis Management – Rescuer*¹. The project goal is to use crowd-sourcing data (image, video, and text captured with mobile devices) to assist in rescue missions. This paper describes the evaluation of techniques to detect fire in image data, one of

¹<http://www.rescuer-project.org/>

the project targets. We use real images from Flickr², a well-known social media website from where we collected a large set of images that were manually annotated as having fire or not fire. We used this dataset as a ground-truth to evaluate image descriptors in the task of detecting fire. Our main contributions are the following:

1. **Curation of the Flickr-Fire Dataset:** a vast human-annotated dataset of real images suitable as ground-truth for the development of content-based techniques for fire detection;
2. **Development of FFireDt:** we propose the Fast-Fire Detection and Retrieval (*FFireDt*), a scalable and accurate architecture for automatic fire-detection, designed over image descriptors able to detect fire, which achieves a precision comparable to that of human annotation;
3. **Evaluation:** we soundly compare the precision and performance of several image descriptors for image classification and retrieval.

Our results provide a solid basis to choose the most adequate pair feature extractor and evaluation function in the task of fire detection, as well as a comprehensive discussion of how the many existing alternatives work in such task.

The remaining of this paper is structured as follows: Section 2 presents the related work; Section 3 presents the main concepts regarding fire-detection in images; Section 4 presents the methodology. Section 5 describes the experiments and discusses their results; finally, Section 6 presents the conclusions.

2 RELATED WORK

Previous efforts on mining information from sets of images include detecting social events and tracking the corresponding related topics which can even include the identification of touristic attractions (Tamura et al., 2012).

Distinctly, in this paper we are interested in the following problem: *Given a collection of photos, possibly obtained by a social media service, how can we efficiently detect fire?* Interesting approaches related to fire motion analysis on video are not applicable for static images (Chunyu et al., 2010), and most of these approaches were found not to work with satisfactory performance (Celik et al., 2007; Ko et al., 2009; Liu and Ahuja, 2004).

Some of the previous works propose the construction of a particular color model focused on fire, based

on Gaussian differences (Celik et al., 2007) or in spectral characteristics to identify fire, smoke, heat or radiation. The spectral color model has been used along with spatial correlation and a stochastic model to capture fire motion (Liu and Ahuja, 2004). However, such technique requires a set of images and is not suitable for individual images, as it is frequent with social media.

Other studies employ a variation of the combination given by a color model transform plus a classifier. This combination is employed in the work of Dimitropoulos (Dimitropoulos et al., 2014), which represents each frame according to the most prominent texture and shape features. It also combines such representation with spatio-temporal motion features to employ SVM to detect fire in videos. However, this approach is neither scalable nor suitable for fire detection on still images.

On the other hand, the feature extraction methods available in the MPEG-7 Standard have been used for image representation in fast-response systems that deal with large amounts of data (Doeller and Kosch, 2008; Ojala et al., 2002; Tjondronegoro and Chen, 2002). However, to the best of our knowledge, there is no study employing those extractors for fire detection. Moreover, despite these multiple approaches, there is no conclusive work about which image descriptors are suitable to identify fire in images.

3 BACKGROUND

3.1 Content-based model for retrieval and classification

The most usual approach to recover and classify images by content relies on representing them using feature extractor techniques (Guyon et al., 2006). After extraction, the images can be retrieved comparing their feature vectors using an evaluation function, which is usually a metric or a divergence function. Comparison is a required step in image retrieval systems. Also, Instance-Based Learning (IBL) classifiers use the evaluation among the images of a given set to label them regarding their visual content (Bedo et al., 2014; Aha et al., 1991). Such concepts can be formalized by the following definitions.

Definition 1. Feature Extraction Method (FEM): A feature extraction method is a non-bijective function that, given an image domain \mathbb{I} , is able to represent any image $i_q \in \mathbb{I}$ in a domain \mathbb{F} as f_q . Each value f_q is called a feature vector (FV) and represents characteristics of an image i_q .

²<https://www.flickr.com/>

In this paper, we use FEMs to represent images in multidimensional domains. Therefore, the image feature vectors can be compared according to the next definition:

Definition 2. Evaluation Function (EF): *Given the feature vectors f_i , f_j and $f_k \in \mathbb{F}$, an evaluation function $\delta : \mathbb{F} \times \mathbb{F} \rightarrow \mathbb{R}$ is able to compare any two elements from \mathbb{F} . The EF is said to be a metric distance function if it complies with the following properties:*

- *Symmetry:* $\delta(f_i, f_j) = \delta(f_j, f_i)$.
- *Non-negativity:* $0 < \delta(f_i, f_j) < \infty$.
- *Triangular inequality:* $\delta(f_i, f_j) \leq \delta(f_i, f_k) + \delta(f_k, f_j)$.

The FEM defines the element distribution in the multidimensional space. On the other hand, the evaluation function defines the behavior of the searching functionalities. Therefore, the combination of FEM and EF is the main parameter to improve or decrease accuracy and quality for both classification and retrieval. Formally, this association can be defined as:

Definition 3. Image Descriptor (ID): *An image descriptor is a pair $\langle \epsilon, \delta \rangle$, where ϵ is a (composition of) FEM and δ is a (weighted) EF.*

By employing a suitable *image descriptor*, it is possible to inspect the neighborhood of a given element considering previous labeled cases. This course of action is the principle of the Instance-Based Learning algorithms, which rely on previously labeled data to classify new elements according to their nearest neighbors. The sense of what “nearest” means is provided by the EF. Formally, this operation must respect the definition bellow:

Definition 4. k Nearest-Neighbors - kNN: *Given an image i_q represented as $f_q \in \mathbb{F}$, an image descriptor $ID = \langle \epsilon, \delta \rangle$, a number of neighbors $k \in \mathbb{N}$ and a set F of images, the k -Nearest Neighbors set is the subset of $F \subset \mathbb{F}$ such that $kNN = \{f_n \in \mathbb{F} \mid \forall f_i \in F; \delta(f_n, f_q) < \delta(f_i, f_q)\}$.*

Once the kNN performance relies on the capability of the *image descriptor* to define the image representations and the search space, it becomes the critical point to be defined in an image retrieval system. Following we review, experiment, and report on multiple possibilities of image descriptors for fire detection.

3.2 MPEG-7 Feature Extraction Methods

The MPEG-7 standard was proposed by the ISO/IEC JTC1 (IEEE MultiMedia, 2002). It defines expected representations for images regarding color, texture and shape. The set of proposed feature extraction

methods were designed to process the original image as fast as possible, without taking into account specific image domains. The original proposal of MPEG-7 is composed of two parts: high and low-level values, both intended to represent the image. The low-level value is the representation of the original data by a FEM. On the other hand, the high-level feature requires examination by an expert.

The goal of MPEG-7 is to standardize the representation of *streamed* or stored images. The low-level FEMs are widely employed to compare and to filter data, based purely on content. These FEMs are meaningful in the context of various applications according to several studies (Doeller and Kosch, 2008; Tjondronegoro and Chen, 2002). They are also supposed to define objects by including color patches, shapes or textures. The MPEG-7 standard defines the following set of low-level extractors (Sato et al., 2010):

- **Color:** Color Layout, Color Structure, Scalable Color and Color Temperature, Dominant Color, Color Correlogram, Group-of-Frames;
- **Texture:** Edge Histogram, Texture Browsing, Homogeneous Texture;
- **Shape:** Contour Shape, Shape Spectrum, Region Shape;

We highlight that shape FEMs’ usually depends of previous object definitions. As the goal of this study relies on defining the best setting for automatic classification and retrieval for fire detection, user interaction on the extraction process is not suitable for our proposal. Thus, we focus only on the color and texture extractors. In this study we employ the following MPEG-7 extractors: Color Layout, Color Structure, Scalable Color, Color Temperature, Edge Histogram, and Texture Browsing. They are explained in the next Sections.

3.2.1 Color Layout

The MPEG-7 Color Layout (CL) (Kasutani and Yamada, 2001) describes the image color distribution considering spatial location. It splits the image in squared sub-regions (the number of sub regions is a parameter) and label each square with the average color of the region. Figure 1(b) depicts the regions for Figure 1(a) according to the Color Layout extractor. Next, the average colors are transformed to the YCbCr space and a Discrete Cosine Transformation is applied over each band of the YCbCr region values. The low-frequency coefficients are extracted through a zig-zag image reading. In order to reduce dimensionality, only the most prominent frequencies are employed in the feature vector.



Figure 1: (a) Original image (b) Regions considered by Color Layout.

3.2.2 Scalable Color

The MPEG-7 Scalable Color (SC) (Manjunath et al., 2001) aims at capturing the prominent color distribution. It is based on four stages. The first stage converts all pixels from the RGB color-space to the HSV space and a normalized color histogram is constructed. The color histogram is quantized using 256 levels of the HSV space. Finally, a Haar wavelet transformation is applied over the resulting histogram (Ojala et al., 2002).

3.2.3 Color Structure

The MPEG-7 Color Structure (CS) expresses both spatial and color distribution (Sikora, 2001). This paper splits the original image in a set of color structures with fixed-size windows. Each fixed-size window selects equally spaced pixels to represent the local color structure, as depicted in Figure 2(a).

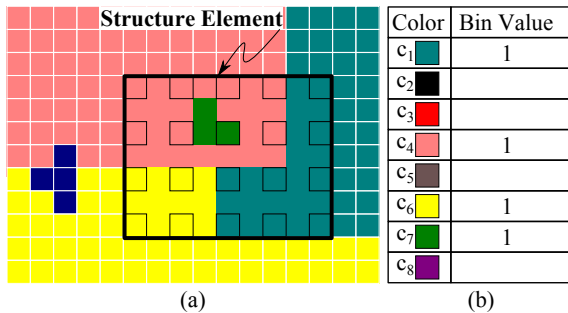


Figure 2: (a) Color structure with a defined window (b) Local histogram.

The window size and the number of local structures are parameters of CS (Manjunath et al., 2001). For each color structure, a quantization based on the HMMD - a color-space derived from HSV that represents color differences - is executed. Then a local "histogram" based on HMMD is built. It stores the presence or absence of the quantized color instead of its distribution along with the window (Figure 2(b)). The resulting feature vector is the accumulated distribution of the local histograms according to the previ-

ous quantization.

3.2.4 Edge Histogram

The MPEG-7 Edge Histogram (EH) aims at capturing local and global edges. It defines five types of edges (Figure 3) regarding $N \times N$ blocks, where N is a extractor parameter. Each block is constructed by partitioning the original image into squared regions.

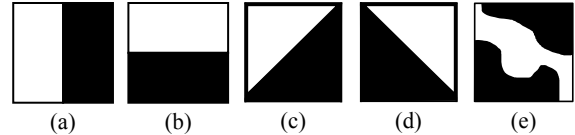


Figure 3: Edge types: (a) Vertical (b) Horizontal (c) 45 degree (d) 135 degree (e) non-directional.

After applying the masks shown in Figure 3 to an image, it is possible to compute the local edge histograms. At this stage, the entire histogram is composed of $5 \times N$ bins, but it is biased by local edges. To circumvent this problem, a variation (Park et al., 2000) was proposed to capture also semi-local edges. Figure 4 illustrates how the 13 semi-local edges are calculated. The horizontal semi-local edges are evaluated first, then the vertical ones and finally the five combinations of the super block edges.

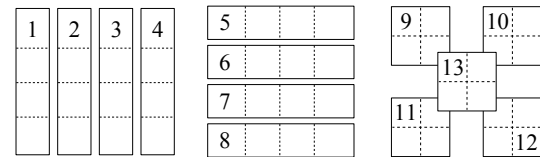


Figure 4: 13 regions corresponding to semi-local edge histograms.

The resulting feature vector is composed of N plus thirteen edge-histograms, which represents the local and the semi-local distribution, respectively.

3.2.5 Color Temperature

The main hypothesis supporting the MPEG-7 Color Temperature (CT) is that there is a correlation between the "feeling of image temperature" and illumination properties. Formally, the proposal considers a theoretical object called *black body*, whereupon its color depends on the temperature (Wnukowicz and Skarbek, 2003). Figure 5 depicts the locus of the theoretical black body, according to Planck formula changing from 2000 Kelvin (red) to 25000 Kelvin (blue).

The feature vector represent the linearized pixels in the XYZ space. This is performed by interactively discarding every pixel with luminance Y above the

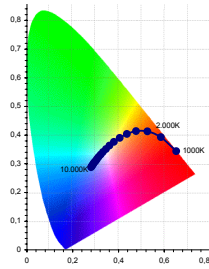


Figure 5: CIE color system and black body locus indicated by the in-out points.

given threshold – a FEM’s parameter. Thereafter, the average color coordinate in XYZ is converted to UCS. Finally, the two closest isotherm lines is calculated from the given color diagrams (Wnukowicz and Skarbek, 2003). The formula for the resulting color temperature depends on the average point, the closest isolines and the distances among them.

3.2.6 Texture Browsing

The MPEG-7 Texture Browsing extractor (TB) is obtained from Gabor filters applied to the image (Lee and Chen, 2005). This FEM parameters’ are the same used in Gabor filtering. Figure 6 (b) illustrates the result of using the Gabor filter to process image 6 (a) following a particular setting. The Texture Browsing feature vector is composed of 12 positions: 2 to represent regularity, 6 for directionality and 4 for coarseness.



Figure 6: (a) Original Image (b) Gabor filter with a kernel setting.

The regularity features represent the degree of regularity of the texture structure as a more/less regular pattern, in such a way that the more regular a texture, the more robust the representation of the other features is. The directionality defines the most dominant texture orientation. This feature is obtained providing an orientation variation for the Gabor filters. Finally, the coarseness represents the two dominant scales of the texture.

3.3 Evaluation Functions

An Evaluation Function expresses the proximity between two feature vectors. We are interested in feature extractor that generates the same amount of features for each image, thus in this paper we account only for evaluation functions for multidimensional spaces. Particularly, we employed distance functions (metrics) and divergences as evaluation functions. Suppose two feature vectors $X = \{x_1, x_2, \dots, x_n\}$ and $Y = \{y_1, y_2, \dots, y_n\}$ of dimensionality n . Table 1 shows the EFs implemented, according to their evaluation formulas.

Table 1: Evaluation functions: their classification as metric distance functions and respective formulas.

Name	Metric	Formula
City-Block	Yes	$\sum_{i=1}^n x_i - y_i $
Euclidean	Yes	$\sqrt{\sum_{i=1}^n (x_i - y_i)^2}$
Chebyshev	Yes	$\lim_{p \rightarrow \infty} (\sum_{i=1}^n x_i - y_i ^p)^{\frac{1}{p}}$
Canberra	Yes	$\sum_{i=1}^n \frac{ x_i - y_i }{ x_i + y_i }$
Kullback Leibler Divergence	No	$\sum_{i=1}^n x_i \ln(\frac{x_i}{y_i})$
Jeffrey Divergence	No	$\sum_{i=1}^n (x_i - y_i) \ln(\frac{x_i}{y_i})$

The most widely employed metric distance functions are those related to the Minkowski family: the Manhattan, Euclidean and Chebyshev (Zezula et al., 2006). A variation of the Manhattan distance is the Canberra distance that results in distances in the range $[0, 1]$. These four EFs satisfy the properties of Definition 2. Therefore, they are metric distance functions.

However, there are non-metric distance functions that are useful for image classification and retrieval. The Kullback-Leibler Divergence, for instance, does not follow the triangular inequality neither the symmetry properties. A symmetric variation of Kullback-Leibler distance is the Jeffrey Divergence, yet it still is not a metric due to the lack of the triangular inequality compliance.

3.4 Instance-Based Learning - IBL

The main hypothesis for IBL classification is that the unlabeled feature vectors (FV) pertain to the same class of its k Nearest-Neighbors, according to a pre-defined rule. Such classifier relies on three resources:

1. An evaluation function, which evaluates the proximity between two FVs;

2. A classification function, which receives the nearest FVs to classify the unlabeled one – commonly considering the majority of retrieved FVs;
3. A concept description updater, which maintains the record of previous classifications.

Variation of these parts defines different IBL versions. For instance, the IB1 – probably the most widely adopted IBL algorithm – adopts the majority of the retrieved elements as the classification rule and keeps no record of previous classifications.

The *kNN* process is able to solve all steps required by IB1. Moreover, it can be seamlessly integrated to the concept of similarity queries by employing extended-SQL expressions (Bedo et al., 2014). This database-driven approach to solve IB1 may reduce the time to obtain the final classification by orders of magnitude, besides the obvious gains obtained by structuring queried data following the entity-relationship model.

4 PROPOSED METHOD

4.1 Dataset Flickr-Fire

We used the Flickr API³ to download 5,962 images (no duplicates) under the Creative Commons license. The images were retrieved using textual queries such as: “fire car accident”, “criminal fire”, and “house burning”. Figures 7 and 8 illustrate samples of the obtained images, which we named *Flickr-Fire*. Even with queries related to fire, some of the images did not contain visual traces of fire, so each image was manually annotated to define a coherent ground-truth dataset.

To perform the annotation, we asked 7 subjects, all of them aging between 20 and 30 years, familiar with the issue, and non-color-blinded. To each subject it was given a subset with 1,589 images that he/she should annotate as containing or not traces of fire. For images in which the annotations disagreed, we asked a third subject to provide an annotation. The average disagreement was 7.2%.

In order to balance the class distribution of the dataset, we randomly removed images to have 1,000 images containing fire and 1,000 images without fire. We made the dataset available online⁴ aiming at the reproducibility of our experiments.

³The Flickr API is available at: www.flickr.com/services/api/

⁴The Flickr-Fire dataset at: www.gbdi.icmc.usp.br



Figure 7: Sample images labeled as 'fire' from dataset Flickr-Fire.



Figure 8: Sample images labeled as 'not-fire' from dataset Flickr-Fire.

4.2 The Architecture of Fast-Fire Detection

Table 2: Feature Extractor Method acronyms used in the experiments.

Feature Extractor Method	Acronym
Color Layout	CL
Scalable Color	SC
Color Structure	CS
Color Temperature	CT
Edge Histogram	EH
Texture Browsing	TB

Table 3: Evaluation Function acronyms used in the experiments.

Evaluation Function Name	Acronym
City-Block	CB
Euclidean	EU
Chebyshev	CH
Canberra	CA
Kullback Leibler Divergence	KU
Jeffrey Divergence	JF

Here we introduce the Fast-Fire Detection (*FFireDt*) architecture, which uses image descriptors

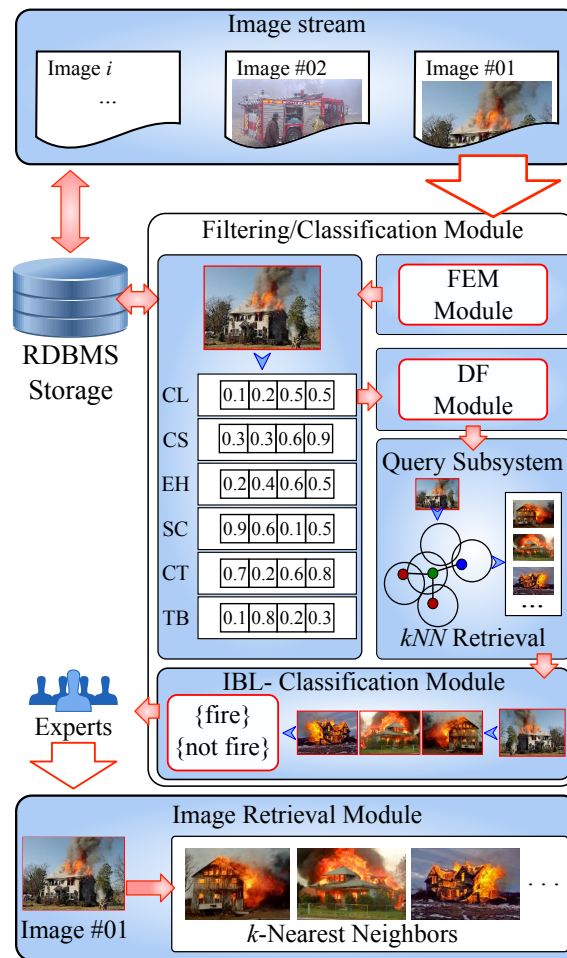


Figure 9: Architecture of the *FFireDt*. The Evaluating Module receives an unlabeled image, represents it executing feature extractor methods and labels it by using the Instance Based Learning Module. The system output (image plus label) interacts with the experts, who may also perform a similarity query.

for image retrieval and classification. The architecture is organized in modules that implement the concepts reviewed in Section 3. Figure 9 illustrates the relationship among the modules, their communication and how they relate to a relational database management system (RDBMS).

The feature extraction methods module (the FEM module) accepts any kind of feature extractor. For this work, we implemented six extractors following the MPEG-7 standard: Color Layout, Scalable Color, Color Structure, Edge Histogram, Color Temperature, and Texture Browsing – explained in Section 3.2. The evaluation functions module (the EF module) is also prepared for general implementations; for this work, we implemented six functions: City-Block, Euclidean, Chebyshev, Jeffrey Divergence, Kullback-Leibler Divergence, and Canberra. The feature extractors methods and evaluation functions acronyms are listed in Tables 2 and 3.

The architecture also has an Instance-Based Learning module (the IBL module), which classifies images labeling them as in classes {fire, not fire}. The IBL module receives as input the unclassified images, one image descriptor (a pair of feature extractor and evaluation function) and the set of past cases correctly labeled. This module is assisted by a similarity retrieval subsystem, which executes the kNN queries necessary for the instance-based learning.

We assume a flow of images is feed to Fast-Fire Detection architecture. As each image arrives, it is stored in the RDBMS along with the corresponding vectors of the features extracted. Then, the IBL module classifies each unlabeled image based on the required descriptor.

This architecture was implemented as an API integrated to an RDBMS, in which the user can create his/her own image descriptor by combining FEM and

DF to perform image classification. The user employs an SQL extension as the front-end for the architecture. The extension is able to execute similarity retrieval (through the Image Retrieval Module) and classification.

5 EXPERIMENTS

In this section, we search the combination of classifiers and image descriptors that are the most suitable to *FFireDt* in the fire detection task. We evaluate the impact of the image descriptors creating a candidate set of 36 descriptors given by the combination of the 6 feature extractors with the 6 evaluation functions considered in this work executing the IB1 classifier over the *Flickr-Fire* dataset. The experiments were performed using the following procedure:

1. Calculate the F-measure metric to evaluate the efficacy of the experimental setting;
2. Select the top-six image descriptors according to the F-measure to generate Precision-Recall plots, bringing more details about the behavior of the techniques;
3. Validate our partial findings using Principal Component Analysis to plot the feature vectors of the extractors;
4. Employed the top-three image descriptors according to the previous measures to perform a ROC curve evaluation, providing the analysis about the most accurate *FFireDt* setting;
5. Evaluate the efficiency of the proposed *FFireDt* architecture, measuring the wall-clock time considering the multiple configurations of the descriptors.

5.1 Obtaining the F-measure

To determine the most suitable *FFireDt* setting, we employed the F-measure, which relies on measuring the number of true positives (TP), false positives (FP) and false negatives (FN). The TP are the images containing fire which are correctly labeled as 'fire', while the FN are those labeled as 'not fire' although being fire images. The FP are the images labeled as 'fire' but containing no traces of fire. The F-measure is given by $2 * TP / (2 * TP + FP + FN)$.

We calculated the F-measure for the 36 image descriptors using 10-fold cross validation. That is, for each round of evaluation, we used one tenth of the dataset to train the IB1 classifier and the remaining data for tests. It is performed 10 times and then the average F-measure is calculated.

Table 4 presents the F-measure values for all the 36 combinations of feature extractor/evaluation function. The highest values obtained for each row are highlighted in bold. The experiment revealed that distinct descriptor combinations impact on fire detection. More specifically, the accuracy of extractors based on color is better than that of the extractors based on texture (Edge Histogram, and Texture Browsing). Moreover, the extractors Color Layout and Color Structure have shown the best efficacy for fire detection, in combination respectively with the evaluation functions Euclidean and Jeffrey Divergence.

The highlighted values are pointed out as the best setting for tuning *FFireDt*. In addition, notice that the best descriptor achieved an accuracy of 85%, which is close to the human labeling process, whose accuracy was 92.8%.

Table 4: F-Measure for each pair of feature extractor method (rows) versus evaluation function (columns). For each feature extractor, the evaluation function with the highest F-Measure is highlighted.

FEM	Evaluation Functions					
	CB	EU	CH	CA	KU	JF
CL	0.834	0.847	0.807	0.828	0.803	0.844
SC	0.843	0.827	0.811	0.835	0.671	0.798
CS	0.853	0.849	0.821	0.848	0.746	0.866
CT	0.799	0.798	0.798	0.800	0.734	0.799
EH	0.808	0.806	0.795	0.806	0.462	0.815
TB	0.766	0.762	0.745	0.751	0.571	0.755

We also compared the best combination achieved by the IB1 classifier, as reported in Table 4, with other classifiers. This was performed to assure that the instance-based learning (the *FFireDt* approach) is the most adequate classification strategy. In these experiments, we tuned *FFireDt* to employ the best EF for each FEM, as reported in Table 4. We also grouped the results according to the employed FEM.

Table 5 shows the *FFireDt* results compared to Naive-Bayes, J48, and RandomForest classifiers. The results show that *FFireDt* achieved the best F-Measure in every but one, of the classification configurations. Random Forest classification using the Scalable Color extractor beat *FFireDt*, although by a narrow F-Measure margin. Thus, we can conclude that IB1 is adequate to fulfill the classification purpose on *FFireDt*.

5.2 Precision-Recall

We extended the analysis of Section 5.1 measuring the Precision and Recall of each configuration, which are values also employed on F-measure. Such

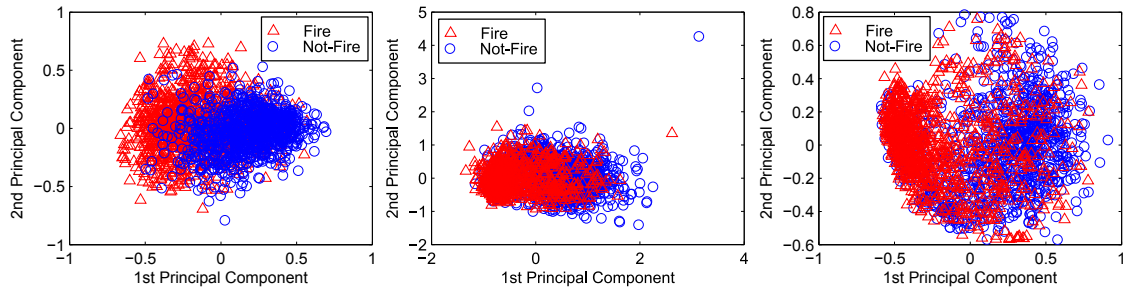


Figure 10: PCA projection of fire and not-fire images: (a) Color Layout, (b) Color Structure, (c) Scalable Color and (d) Edge Histogram. The Color Layout visually separates the dataset into two clusters.

Table 5: *FFireDt* obtained the highest F-Measure for all but one FEM when compared to other classifiers. For each feature extractor, we highlighted strategy with the highest F-Measure.

FEM	Classifiers			
	<i>FFireDt</i>	Naive-Bayes	J48	Random Forest
CL	0.847	0.787	0.751	0.829
SC	0.843	0.808	0.845	0.864
CS	0.866	0.406	0.842	0.866
CT	0.800	0.341	0.800	0.774
EH	0.815	0.522	0.711	0.787
TB	0.766	0.476	0.706	0.723

further analysis permits to better understand the behavior of the image descriptors and more specifically, the behavior of the feature extractors. A Precision vs. Recall ($P \times R$) curve is suitable to measure the number of relevant images regarding the number of retrieved elements. We used Precision vs. Recall as a complementary measure to determine the potential of each image descriptor in the *FFireDt* setting. A rule of thumb on reading $P \times R$ curves is: the closer to the top the better the result is. Accordingly, we consider

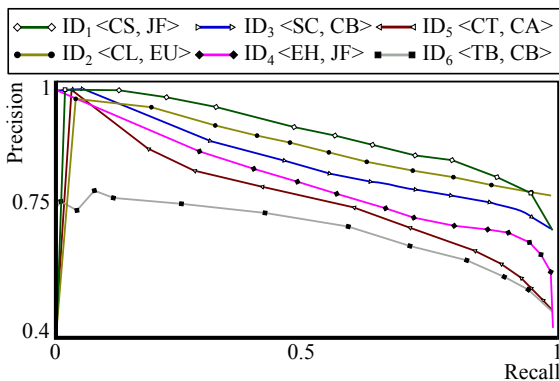


Figure 11: Precision vs. Recall graphs for each of the most precise image descriptor combinations according to F-measure.

only the more efficient combination of each feature extractor, as highlighted in Table 4: $ID_1 <CS, JF>$, $ID_2 <CL, EU>$, $ID_3 <SC, CB>$, $ID_4 <EH, JF>$, $ID_5 <CT, CA>$ and $ID_6 <TB, CB>$. Figure 11 confirms that the image descriptors ID_1 , ID_2 , and ID_3 are in fact the most effective combinations for fire detection. It also shows that, for those three descriptors, the precision is at least 0.8 for a recall of up to 0.5, dropping almost linearly with a small slope, which can be considered acceptable. This observation reinforces the findings of the F-measure metric, indicating that the behavior of the descriptors are homogeneous and well-suited for the task of retrieval and, consequently, for classification purposes.

5.3 Visualization of feature extractors

Based on the results shown so far, we hypothesize that the Color Structure, Color Layout, and Scalable Color extractors are the most adequate to act as *FFireDt* setting. In this section, we look for further evidence, using visualization techniques to better understand the feature space of the extractors using Principal Component Analysis (PCA). The PCA analysis takes as input the extracted features, which may have several dimensions according to the FEM domain, and reduces them to two features. Such reduction allows us to visualize the data as a scatter-plot.

Our hypothesis shall gain more credibility if the corresponding visualizations allow seeing a best separation of the classes *fire* and *not-fire*, in comparison to the other three extractors. Figures 10(a) - 10(d) allow visualizing the two-dimensional projection of the data, plotting the two principal components of the PCA processing of the space generated by each extractor. Figure 10(a) depicts the representation of the data space generated by CL, the extractor that presented the better separability in the classification process. The two clusters can be seen as two well-formed clouds with a reasonably small overlapping, splitting the images as containing fire or not.

Figure 10(b) shows the data visualization of the

space generated by CS, which was the FEM that obtained the highest F-measure on previous experiments. The data projection shows that each cluster forms a cloud clearly identifiable, having the centers of the clouds distinctly separated. However, this figure reveals that there is a large overlap between the two classes. Figure 10(c) presents the projection of the space generated by the SC extractor. Again, it can be seen that there are two clusters, but with an even larger overlap between them. Visually, the CL outperformed the other color FEMS: the CS and SC, when drawing the border between the two classes.

Figure 10(d) depicts the visualization generated by the EH extractor. It can be seen that indeed it has two clouds: one almost vertical to the left and another along the “diagonal” of the figure. However, the two clouds are not related to the existence of fire, as the elements of both clusters are distributed over both clouds. Figures 10 (a), 10 (b) and 10 (c) show the visualization of the extractors based on color. The four visualizations show that the corresponding CL, SC and SC indeed generate clusters. However, there are increasing larger overlaps between fire and not-fire instances. Regarding the TB and CT features, the PCA projection was not able to separate the fire and not-fire classes. Concluding, the visualization of the feature spaces shows that extractors based on color are able to separate the data into visual clouds related to the expected clusters. Particularly, Color Layout has shown the best visualization, followed by Color Structure, and Scalable Color, which also have shown to significantly separate the classes. However, the extractors based on texture identify characteristics that are not related to fire, thus presenting the worst separability, as expected.

5.4 ROC curves

Finally, we employed one last accuracy measure to define the *FFireDt* setting: the ROC curve. It allows us to determine the experiments overall accuracy, using measures of sensitivity and specificity. Figure 12 presents the detailed ROC curves for image descriptors $ID_1 = \langle CS, JF \rangle$, $ID_2 = \langle CL, EU \rangle$, and $ID_3 = \langle SC, CB \rangle$, the top three best combinations according to the F-Measure, Precision-Recall and Visualization experiments. For fire-detection, the area under the ROC curve was up to 0.93 for ID_1 ; up to 0.87 for ID_2 ; and up to 0.85 for ID_3 .

These results indicate that the top three image descriptors have similar and satisfactory accuracy. Therefore, the choice of which descriptor to use becomes a matter of performance. In the next section we evaluate the performance to conclude what is the

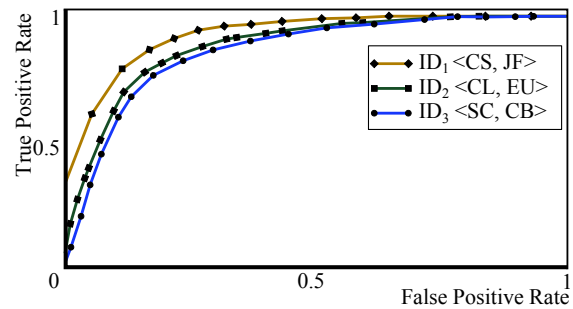


Figure 12: ROC curves for the top three image descriptors in the task of fire detection.

most adequate descriptor.

5.5 Processing Time and Scalability

When monitoring images originated from social media, the time constraint is important because of the high rate at which new images arrive. Thus, we also evaluate the efficiency, given in wall-clock time, of the candidate image descriptors. We ran the experiments in a personal computer equipped with processor Intel Core i7 R 2.67 GHz with 4GB memory over operating system Ubuntu 14.04 LTS.

Feature Extractors

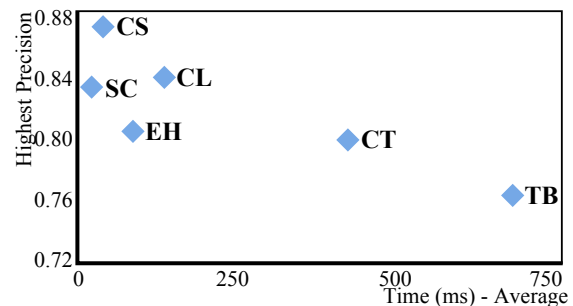


Figure 13: Plot *Highest precision when classifying dataset Flickr-Fire* vs. *Average time to extract the features of one image* for the six feature extractors.

When monitoring images originated from social media, the time constraint is important because of the high rate at which new images may arrive. Thus, we also evaluate the efficiency, given in wall-clock time, of the candidate image descriptors. Figure 13 shows the required average time to perform the feature extraction on *FFireDt* regarding *Flickr-Fire* dataset. Color Structure, the most precise extractor, was the second fastest. The second and third most precise extractors were Color Layout and Scalable Color: the former was three times slower than Color Structure, and the later was the fastest extractor. Thus, we are

now able to state that the extractors Color Structure and Scalable Color are the best choices for fire detection in image streams. Meanwhile, the texture-based extractors Edge Histogram, and Texture Browsing presented low performances, so they are definitely dismissed as possible choices.

Evaluation Functions

Figure 14 shows the time required to perform 2 trillion evaluation calculations for each evaluation function on feature vectors of 256 dimensions. The plot *average precision vs. wall-clock time* shows that, although the Jeffrey Divergence demonstrated the highest precision, it was the least efficient. In their turn, the City-Block and Euclidean distances presented excellent performance and a precision only slightly below the Jeffrey Divergence. Therefore, we can say that they are the most adequate evaluation functions when performance is on concern, such as is the case in our problem domain. Finally, we conclude that the image descriptors given by the combinations $\{CS, SC\} \times \{CB, EU\}$ are the best options in terms of both efficacy (precision) and efficiency (wall-clock time). In Table 6 we reproduce the F-measure results highlighting the most adequate combinations according to our findings.

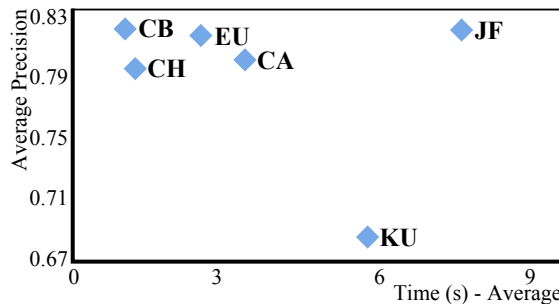


Figure 14: Plot *Average precision when classifying dataset Flickr-Fire vs. Time to perform 2 trillion calculations* for the six evaluation functions.

Table 6: F-Measure for each pair of feature extractor method (rows) and evaluation function (columns), now highlighting the best combinations according to our experiments.

FEM	Evaluation Functions					
	CB	EU	CH	JF	KU	CA
CL	0.834	0.847	0.807	0.844	0.803	0.828
CS	0.853	0.849	0.821	0.866	0.746	0.848
SC	0.843	0.827	0.811	0.798	0.671	0.835
EH	0.808	0.806	0.795	0.815	0.462	0.806
CT	0.799	0.798	0.798	0.799	0.734	0.800
TB	0.766	0.762	0.745	0.755	0.571	0.751

By these experiments, we point out that the best

image descriptor for fire detection, considering the built dataset, is given by the combinations of the MPEG-7 Color Structure and Scalable Color extractors with the distance functions City-Block and Euclidean. These combinations provide not only more efficacy, but also more efficiency. In general, we noticed that feature extractors based on color were more effective than extractors based on texture. We also identified that the Jeffrey divergence was the most accurate, however, it was also the most expensive evaluation function.

6 CONCLUSIONS

We worked on the problem of identifying fire in social-media image sets in order to assist rescue services during emergency situations. The approach was based on an architecture for content-based image retrieval and classification. Using this architecture, we compared the accuracy and performance (processing time) of image descriptors (pairs of feature extractor and evaluation function) in the task of identifying fire in the images. As a ground-truth, we built a dataset with 2,000 human-annotated images obtained from website Flickr. Our contributions in this paper are summarized as follows:

1. **Dataset Flickr-Fire:** we built a varied human-annotated dataset of real images suitable as ground-truth to foster the development of more precise techniques for automatic identification of fire;
2. **Fast-Fire Detection (FFireDt) Architecture:** we designed and implemented a flexible, scalable, and accurate method for content-based image retrieval and classification to be used as a model for future developments in the field;
3. **Evaluation of existing techniques:** we compared 36 combinations of MPEG-7 feature extractors with evaluation functions from the literature considering their potential for accurate retrieval using F-measure and Precision-Recall. As a result, we achieved 85% accuracy, precise classification (ROC curves), and efficient performance (wall-clock time).

Our results showed that *FFireDt* was able to achieve a precision for the fire detection task, which is comparable to that of human annotators. We conclude by stressing the importance of monitoring images from social media, including situations in which decision making can benefit from precise and accurate information. However, to take advantage of them, it is required to have automated tools that can flag them

from the social media as soon as possible, and this work is an important step toward it.

ACKNOWLEDGEMENTS

This research is supported, in part, by FAPESP, CNPq, CAPES, STIC-AmSud, the RESCUER project, funded by the European Commission (Grant: 614154) and by the CNPq/MCTI (Grant: 490084/2013-3).

REFERENCES

- Aha, D. W., Kibler, D., and Albert, M. K. (1991). Instance-based learning algorithms. *Mach. Learn.*, 6(1):37–66.
- Bedo, M. V. N., Traina, A. J. M., and Traina Jr., C. (2014). Seamless integration of distance functions and feature vectors for similarity-queries processing. *JIDM*, 5(3):308–320.
- Celik, T., Demirel, H., Ozkaramanli, H., and Uyguroglu, M. (2007). Fire detection using statistical color model in video sequences. *Journal of Visual Communication and Image Representation*, 18(2):176 – 185.
- Chunyu, Y., Jun, F., Jinjun, W., and Yongming, Z. (2010). Video fire smoke detection using motion and color features. *Fire Technology*, 46(3):651–663.
- Dimitropoulos, K., Barmpoutis, P., and Grammalidis, N. (2014). Spatio-temporal flame modeling and dynamic texture analysis for automatic video-based fire detection. *Circuits and Systems for Video Technology, IEEE Transactions on*, PP(99):1–1.
- Doeller, M. and Kosch, H. (2008). The mpeg-7 multimedia database system (mpeg-7 mddb). *Journal of Systems and Software*, 81(9):1559 – 1580.
- Guyon, I., Gunn, S., Nikravesh, M., and Zadeh, L. A. (2006). *Feature Extraction: Foundations and Applications (Studies in Fuzziness and Soft Computing)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.
- IEEE MultiMedia (2002). Mpeg-7: The generic multimedia content description standard, part 1. *IEEE MultiMedia*, 9(2):78–87.
- Kasutani, E. and Yamada, A. (2001). The mpeg-7 color layout descriptor: a compact image feature description for high-speed image/video segment retrieval. In *Proceedings of the International Conference on Image Processing*, volume 1, pages 674–677 vol.1.
- Ko, B. C., Cheong, K.-H., and Nam, J.-Y. (2009). Fire detection based on vision sensor and support vector machines. *Fire Safety Journal*, 44(3):322 – 329.
- Kudyba, S. (2014). *Big Data, Mining, and Analytics: Components of Strategic Decision Making*. Taylor & Francis Group.
- Lee, K.-L. and Chen, L.-H. (2005). An efficient computation method for the texture browsing descriptor of mpeg-7. *Image Vision Comput.*, 23(5):479–489.
- Liu, C.-B. and Ahuja, N. (2004). Vision based fire detection. In *Proceedings of the 17th International Conference on Pattern Recognition.*, volume 4, pages 134–137 Vol.4.
- Manjunath, B. S., Ohm, J. R., Vasudevan, V. V., and Yamada, A. (2001). Color and texture descriptors. *IEEE Trans. Cir. and Sys. for Video Technol.*, 11(6):703–715.
- Ojala, T., Aittola, M., and Matinmikko, E. (2002). Empirical evaluation of mpeg-7 xm color descriptors in content-based retrieval of semantic image categories. In *Proceedings of 16th International Conference on Pattern Recognition*, volume 2, pages 1021–1024 vol.2.
- Park, D. K., Jeon, Y. S., and Won, C. S. (2000). Efficient use of local edge histogram descriptor. In *Proceedings of the ACM Workshops on Multimedia*, pages 51–54. ACM.
- Russo, M. R. (2013). Emergency management professional development: Linking information communication technology and social communication skills to enhance a sense of community and social justice in the 21st century. In *Crisis Management: Concepts, Methodologies, Tools and Applications*, pages 651–663. IGI Global.
- Sato, M., Gutu, D., and Horita, Y. (2010). A new image quality assessment model based on the MPEG-7 descriptor. In *Advances in Multimedia Information Processing*, pages 159–170. Springer.
- Sikora, T. (2001). The mpeg-7 visual standard for content description-an overview. *IEEE Trans. Cir. and Sys. for Video Technol.*, 11(6):696–702.
- Tamura, S., Tamura, K., Kitakami, H., and Hirahara, K. (2012). Clustering-based burst-detection algorithm for web-image document stream on social media. In *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 703–708. IEEE.
- Tjondronegoro, D. and Chen, Y.-P. (2002). Content-based indexing and retrieval using mpeg-7 and x-query in video data management systems. *World Wide Web*, 5(3):207–227.
- Villela, K., Breiner, K., Nass, C., Mendonça, M., and Vieira, V. (2014). A smart and reliable crowdsourcing solution for emergency and crisis management. In *22nd Interdisciplinary Information Management Talks (IDIMT 2014)*, Podebrady, Czech Republic.
- Wnukowicz, K. and Skarbek, W. (2003). Colour temperature estimation algorithm for digital images - properties and convergence. In *Opto Eletronics Review*, volume 11, pages 193–196.
- ZeZula, P., Amato, G., Dohnal, V., and Batko, M. (2006). *Similarity Search - The Metric Space Approach*, volume 32 of *Advances in Database Systems*. Springer Publishing Company, Incorporated, Berlin, Heidelberg.